

# Principles of Programming Languages Midterm Exam

CS374 Fall 2022

Professor: William M. Mongan

Date: October 3, 2022

Name: \_\_\_\_\_

Student ID: \_\_\_\_\_

**Instructions:** The exam must be completed in 50 minutes. You should read all of the questions before you begin and plan your time accordingly. The exam is worth **100 points**. Not all questions have the same point value and not all questions should take the same amount of time.

The exam is open-book and you may use your class notes. **There will be absolutely no access to any wireless or wired network during this exam outside of that required to open and edit this document or the class resources and notes.**

You should write your answers in the space provided. Try to make your answers as clear and concise as possible, and write them legibly. Show all your work as partial credit may be given.

1. **[10 points]** Suppose we have a C++ class structure

```
class Vehicle {
    public:
        void honk() { cout << "Beep!" << endl; }
};

class BigRigTruck: public Vehicle {
    public:
        void honk() { cout << "HONK!" << endl; }
};
```

Suppose we have a variable of declared type Vehicle:

```
Vehicle truck = BigRigTruck();
truck.honk();
```

We wish for the call to honk() to print the inherited function, but because the name is statically bound, we will print "Beep!" as associated with the declared type.

**(2 Points)** What keyword can we add to the honk() definition in Vehicle to invoke the inherited method?

**(5 Points)** Suppose this binding was resolved statically (at compile time), as opposed to dynamically (at run time, immediately when the function is called during execution). If that were the case, what would happen if the method **honk()** were called on an object whose type was cast at runtime from a **Vehicle** to a **BigRigTruck**?

**(3 Points)** With this in mind, when/how do you think these bindings are resolved (statically or dynamically)?

2. **[10 Points]** Suppose you are implementing a compiler and wish to implement polymorphism while maintaining static binding for efficiency. For example, you want to allow two functions such as the ones below, but do not want to keep track of strong typing, and thus you want to be able to put the appropriate function call right into your assembly and machine code at compile time (statically). What is one way in which you could implement this while ensuring that calls to each function result in the appropriate bound function being called?

```
int compare(int x, int y);  
int compare(char* x, char* y);
```

3. **[5 Points]** From the compiler's perspective, what does it mean for a method to be **static** in Java?
4. **[10 Points]** If a variable is declared to be static or global, and a variable of the same name is declared as a local function variable, how can a compiler or linker locate the correct binding within the function scope (or, outside of the function, in the global scope)? Specifically, what kind of data structure would help you to do this, and how?

5. [10 Points] Suppose you have a Bash variable DATA which contains the string

Bill,100

Using Bash pipe and filtering, the cut command (or an awk command), and the variable DATA, print the value 100 to the screen. **You must use each of these to receive credit for this question.**

6. [10 Points] Suppose you have a variable in a Bash script called NAME, which contains the value "John Doe". Using Bash variable substitution, echo the word "Johnson" to the screen. To do this, use a variable expansion operator inside the \${} variable operator to remove everything from the first space character onwards, then echo "son" immediately thereafter. You can **(and must)** do this with a single line of Bash code.

7. [10 Points] Suppose you have two lists:  $x$  and  $y$ , and you wish to compute their dot product:

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n - m],$$

In other words, it is the dot product of the list  $\mathbf{f}$  with the reverse of the list  $\mathbf{g}$ .

Write a procedure **dot** to compute the dot product of two lists, and then write a procedure **convolution** that computes the convolution of two lists (which may use **dot**). You may use **map** and **apply**, but you may also write the procedure using a recursive list processing strategy.

You may assume that there is a built-in procedure called **reverse**, which accepts a list as a parameter and returns the reverse of the list.

8. [10 Points] Sketch scheme pseudocode for a hypothetical function **czr** that returns the **last** item in a list. Do not use a **reverse** procedure, unless you write one here yourself.

**9. [10 Points]** Write a Prolog-style program to define people as parents of other people. Using transitive closure, define a Horn Clause that indicates if a person  $X$  is an ancestor of a child  $Y$  (if  $Y$  has any number of parents  $Z_1$  with parent  $Z_k \dots$  eventually having parent  $X$ ). For example, a grandparent, great-grandparent, and parent are all ancestors of a child.

**10. [5 Points]** In your own words, what is a lambda expression?

**11. [10 Points]** Python and scheme are examples of dynamically typed languages.

**(5 Points)** How do these differ from statically typed languages like Java?

**(5 Points)** What is one advantage and one disadvantage of using a dynamically typed language?